

Domotica

Protocol

ICT Automatisering Nederland N.V.			
Project	Domotica	Document ID	N.V.T
Auteur(s)	David Ren	Status	Concept
Reviewer(s)	Billy van der Burgt	Datum	6 juni 2014
	Khalid el Mrabet	Classificatie	Algemeen
Template	ICT2.3003N v2.7	Versie	0.5
Geprint	06/06/14 03:34		
Bestandsnaam	3003N Algemene Document Standaard.doc		

Historie

Versie	Datum	Auteur	Omschrijving
0.1			Initiële versie
0.2	20-03-2014	David Ren	Feedback van Billy van der Burgt verwerkt
0.3	20-03-2014	David Ren	Feedback van Khalid el Mrabet verwerkt
0.4	21-03-2014	David Ren	Feedback van Billy van der Burgt verwerkt
0.5	10-04-2014	David Ren	Reply frame opgesteld, error list, general commands

Distributie

Versie	Datum	Aan	Bedrijf
--------	-------	-----	---------

Dit document is eigendom van ICT Automatisering Nederland B.V. Niets uit deze uitgave mag worden verveelvoudigd en/of openbaar gemaakt door middel van druk, fotokopie, microfilm of op welke wijze ook, zonder voorafgaande schriftelijke toestemming van de eigenaar.

© 2014 ICT Automatisering Nederland B.V. Alle rechten voorbehouden.

Inhoudsopgave

1	Introductie	4
1.1	Doel	4
1.2	Scope	4
1.3	Termen en afkortingen	4
1.4	Referenties	4
1.5	Overzicht	4
2	Protocol	5
2.1	Huidig	5
2.1.1	Modulelijst	5
2.1.2	Takenlijst	5
2.2	Uitbreiding	6
2.2.1	Reply frame en errorlist	6
2.2.2	Modulelijst	6
2.2.3	Takenlijst	6
2.2.3.1	Unknown	6
2.2.3.2	LED	7
2.2.3.3	Temperatuursensor	7
2.2.3.4	Rolluik	7
2.2.3.5	Radio	7
	Bi-directionele communicatie	8
2.3	Sequence diagram	9
2.4	Activity Diagram	10
3	Literatuurlijst	11

1 Introductie

1.1 Doel

De uitgebreide commando's, errorlist, antwoordframe vastleggen in een document.

1.2 Scope

Er wordt geen aanpassing gedaan op de manier waarop de data van de Raspberry Pi naar de module verstuurd wordt.

1.3 Termen en afkortingen

Term	Omschrijving
------	--------------

Afkorting	Omschrijving
-----------	--------------

1.4 Referenties

Referentie	Document ID	Omschrijving
------------	-------------	--------------

1.5 Overzicht

2 Protocol

2.1 Huidig

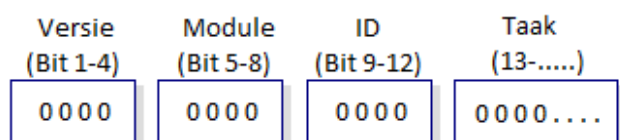
De data wordt door het C programma(op de Raspberry Pi) wordt op de manier als volgt opgebouwd.

Versie: De Versie staat voor de versie van het protocol. Mochten zaken als de te versturen data wijzigen of wordt de manier van verzenden anders dan is het mogelijk om de versie te veranderen.

Module:De Module staat voor het soort apparaat dat aangestuurd zal gaan worden. In dit geval is het alleen nog mogelijk om een led aan te sturen.

ID: De ID staat voor de betreffende led in het netwerk.

Taak: De Taak staat voor de taak die met de betreffende module en ID uitgevoerd moet gaan worden. Op dit moment is het mogelijk om de led aan of uit te zetten, de intensiteit of de kleurcode te veranderen.



Figuur 1 Protocol

2.1.1 Modulelijst

0x01 = LED

2.1.2 Takenlijst

0x01 = Set color

- Kleurcode in hexadecimale getalen
- Bestaat uit 3 waardes(vb. 0x00 0xFF 0x54)

0x02 = Set intensity

- Waarde van 0 tot 100 (percentage)

0x03 = Set state

- 0 = uit
- 1 = aan

2.2 Uitbreiding

2.2.1 Reply frame en errorlist

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
State	Received CMD	Data Error	Data	Data	Data	Data	Data

State (Byte 1)

0x00 = Success

0xFF = Error

Error list (Byte 3)

0x01 = Wrong module

- Byte 4: Sent module type
- Byte 5: Module type

0x02 = Wrong version

- Byte 4: Sent version
- Byte 5: Version

0x03 = Wrong parameter size

- Byte 4: Sent n parameters, n will be sent back
- Byte 5: Expected n parameters

0x04 = Wrong parameter range

- Byte 4: Sent parameter
- Byte 5: Expected parameter minimum range
- Byte 6: Expected parameter maximum range

0x05 = Unknown command

- Byte 4: Sent command

0x06 = Module defect

0x07 = Corrupt Data

2.2.2 Modulelijst

0x01 = LED

0x02 = Temperatuursensor

0x03 = Rolluik

0x04 = Radio

0xFF = General

2.2.3 Takenlijst

2.2.3.1 Unknown

0x01 = Get module type

0x02 = Get version number

2.2.3.2 LED

0x01 = Set color

- Kleurcode in hexadecimale getallen
- Bestaat uit 3 waardes(vb. 0x00 0xFF 0x54)

0x02 = Set intensity

- Waarde van 0 tot 100(percentage)

0x03 = Set state

- 0 = uit
- 1 = aan

0x81 = Get color

- Kleurcode in hexadecimale getallen
- Bestaat uit 3 waardes(vb. 0x00 0xFF 0x54)

0x82 = Get intensity

- Waarde van 0 tot 100(percentage)

0x83 = Get state

- 0 = uit
- 1 = aan

2.2.3.3 Temperatuursensor

0x81 = Get temperature

- Decimale waarde gesplitst in twee
- 0-255 voor getal voor de komma
- 0-9 voor getal na de komma

2.2.3.4 Rollluik

0x01 = Set position

- Waarde van 0 tot 100(percentage)

0x81 = Get position

- Waarde van 0 tot 100(percentage)

2.2.3.5 Radio

0x01 = Set volume

- Waarde van 0 tot 100(percentage)

0x02 = Set channel

- Decimale waarde gesplitst in twee
- 0-255 voor getal voor de komma
- 0-9 voor getal na de komma

0x03 = Mute

- 0 = unmute
- 1 = mute

0x04 = Seek channel

- 0 = seek down
- 1 = seek up

0x81 = Get volume

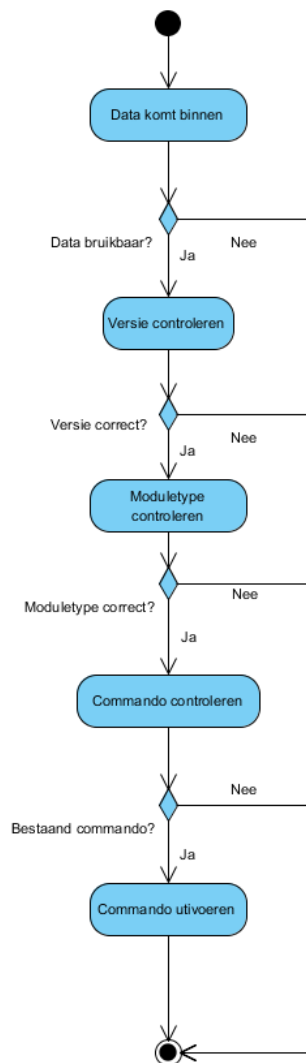
- Waarde van 0 tot 100(percentage)

0x82 = Get channel

- Decimale waarde gesplitst in twee
- 0-255 voor getal voor de komma
- 0-9 voor getal na de komma

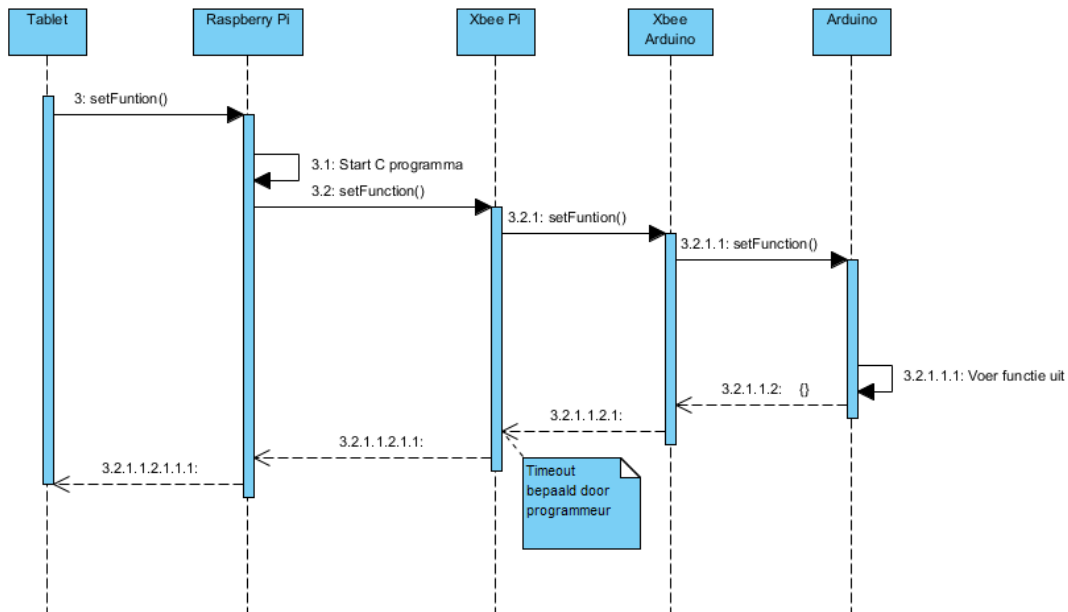
Bi-directionele communicatie

De huidige applicatie kan op het moment alleen maar 1 kant op communiceren, het is de bedoeling dat de applicatie uitgebreid wordt zodat het actuele informatie kan opvragen. Om dit te realiseren moet de code op zowel de frontend als backend uitgebreid worden met get commando's. Hiervan is een sequence diagram gemaakt van de procedure dat gevolgd zal worden en ook een activity diagram voor de aanpaste code op de Arduino.



2.3 Sequence diagram

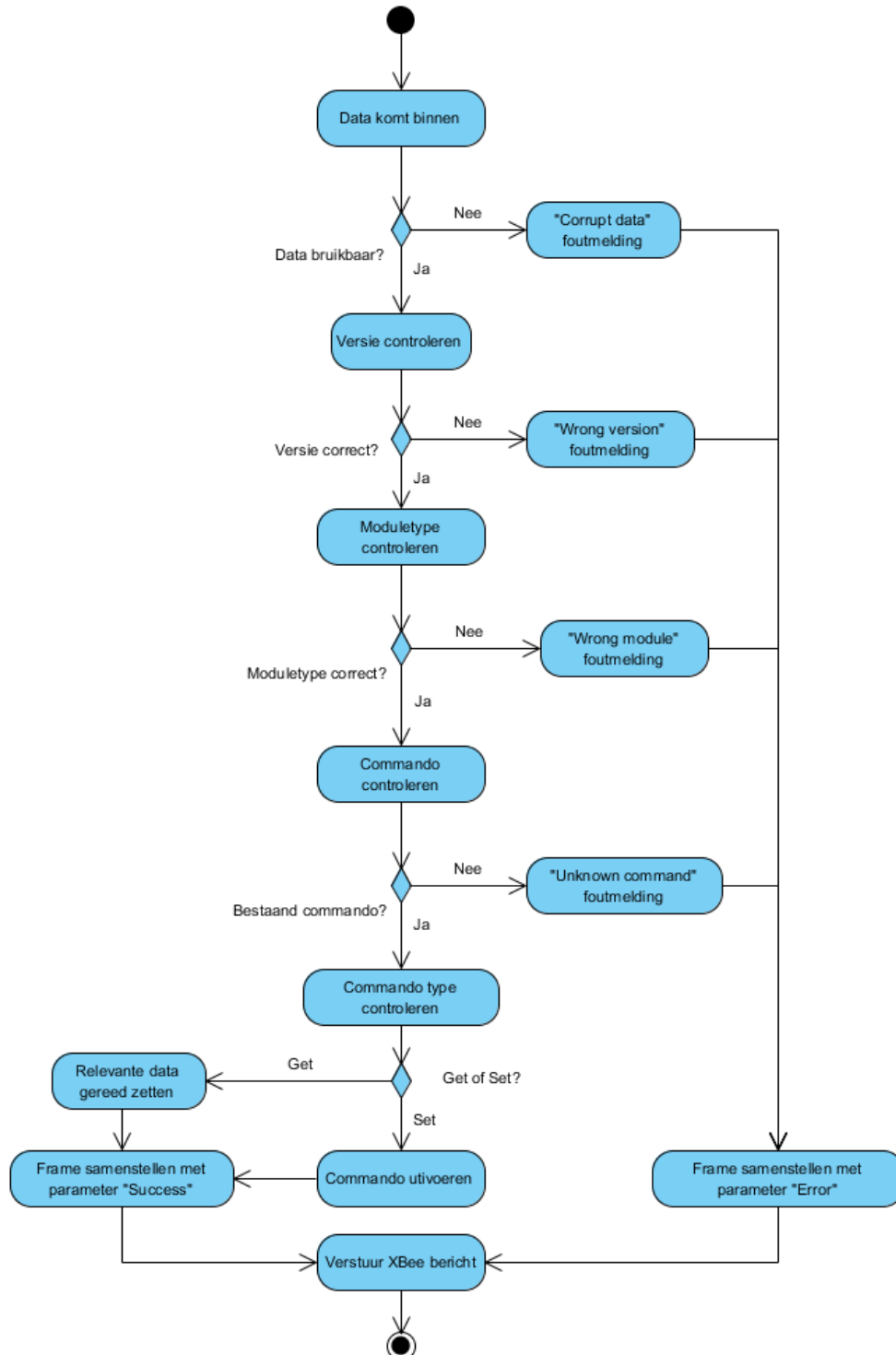
Hieronder staat het sequence diagram voor een get commando vanuit de applicatie naar de Arduino. Hierbij zal de applicatie een commando versturen via de Raspberry Pi naar de Arduino toe en zal dan enkele secondes wachten op een antwoord.



2.4 Activity Diagram

De code op de Arduino zal aangepast moeten worden zodat het data terugstuurt. Zie onderstaande afbeelding voor het huidige diagram:

Dit zal uitgebreid moeten worden naar de onderstaand activity diagram.



3 Literatuurlijst

Algemene documentatie.docx – Jeroen Wennekes